IN THE CLAIMS

Please cancel Claims 33, 34, and 36 without prejudice, and amend Claims 1, 14, 17, 23, and 41 as follows:

1. (Currently Amended) A method of controlling the execution of instructions within a pipelined processor, comprising:

providing an instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said words comprising a jump instruction having at least one user-configurable mode and at least one user-definable mode associated therewith, said user-configurable and user-definable modes each being specified by the same ones of said plurality of bits;

assigning one of a plurality of values to [at least one] said ones of said data bits of said at least one jump instruction; and

controlling the execution of at least one subsequent instruction within said pipeline based on said one assigned value of said [at least one data bit] data bits when said at least one jump instruction is decoded.

2. (Original) The method of Claim 1, wherein the act of assigning comprises:

identifying a plurality of data bits within said at least one jump instruction; and

assigning one of two discrete values to each of said data bits, the combination of said two discrete values representing at least three jump delay slot modes within said processor.

3. (Original) The method of Claim 2, wherein the act of controlling the execution based on said discrete values comprises selecting at least one mode from the group comprising:

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

4. (Original) The method of Claim 3, wherein said at least one jump instruction comprises a conditional branch instruction.

5. (Original) The method of Claim 1, wherein the act of controlling the execution based on said one assigned value comprises:

(i) executing said at least one subsequent instruction under all circumstances;

(ii) executing said at least one subsequent instruction only if a jump occurs; and

(iii) stalling the pipeline or inserting a bubble into the pipeline if a jump occurs.

6.-13. (Cancelled)

14. (Currently Amended) A digital processor comprising:

a processor core having a multistage instruction pipeline, said core being adapted to decode and execute an instruction set comprising a plurality of instruction words;

a data interface between said processor core and an information storage device; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a user-configurable jump instruction containing data defining a plurality of jump delay slot modes and at least one user-defined mode, said jump delay slot modes and at least one user-defined mode each being specified by the same portions of said data, said plurality of modes controlling the execution of instructions within said instruction pipeline of said processor core in response to said at least one jump instruction word within said instruction set.

15. (Original) The processor of Claim 14, wherein said plurality of jump delay slot modes comprises at least the following modes:

(i) executing a subsequent instruction within said pipeline under all circumstances;

(ii) executing a subsequent instruction within said pipeline only if jumping occurs; and

(iii) stalling the pipeline if jumping occurs.

16. (Original) The processor of Claim 14, wherein said at least one jump instruction comprises a conditional branch instruction having an associated logical condition, the execution of a jump to the address within said information storage device specified by said at least one conditional branch instruction being determined by said logical condition.

17. (Currently Amended) A digital processor having at least one pipeline and an associated data storage device, wherein the execution of instructions within said at least one pipeline is controlled by the method comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a user-configurable branch instruction having at least one user-defined mode, said branch instruction directing branching to a first address within said data storage device;

assigning one of a plurality of values to each of said data bits of said at least one branch instruction;

decoding said at least one branch instruction including said [one] assigned values;

determining whether to execute an instruction within said pipeline in a stage preceding that of said at least one branch instruction based at least in part on said one values; [and]

branching to said first address based on said at least one branching instruction; and

performing, based at least in part on said act of decoding, said assigned values, at least one other function dictated by said at least one user-defined mode.

18. (Original) The processor of Claim 17, wherein said data bits comprise binary (base 2) data.

19. (Original) The method of Claim 17, wherein said at least one pipeline comprises at least a three stage instruction pipeline comprising instruction fetch, decode, and execute stages.

20. (Original) A method of controlling program operation of a multi-stage pipelined digital processor, comprising:

storing an instruction set within said data storage device, said instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said instruction words comprising a branch instruction directing branching to a first address within said data storage device based on a first parameter;

defining a plurality of jump delay slot modes comprising;

(i) executing a subsequent instruction under all circumstances;

(ii) executing a subsequent instruction only if jumping occurs;

(iii) stalling the pipeline for one cycle if jumping occurs; and

(iv) stalling the pipeline for two or more cycles if jumping occurs;

assigning at least one of said plurality of jump modes to at least two of said data bits of said at least one branch instruction;

decoding said at least one branch instruction including said at least two data bits; and

controlling said pipeline based at least in part on said at least two data bits and said first parameter.

21-22. (Cancelled)

23. (Currently Amended) A digital processor comprising:

processing means having a multistage data pipeline, said processing means being adapted to decode and execute an instruction set comprising a plurality of instruction words;

means for storing data;

data interface means for transferring data between said processing means and said means for storing data; and

an instruction set comprising a plurality of instruction words, at least one of said instruction words being a user-configurable jump instruction containing data defining a plurality of jump control means and at least one user-defined means, said jump control means and at least one user-defined means each being specified by the same portions of said data, said plurality of jump control means controlling the execution of instructions within said data pipeline of said processing means in response to said at least one jump instruction word within said instruction set.

24. (Cancelled)

25. (Previously Presented) The method of Claim 1, wherein at least one of said plurality of instruction words comprises an op-code and a plurality of fields, each of said fields comprising a plurality of bits, said at least one instruction word being encoded according to the method comprising:

associating a first of said fields with a first data source;

associating a second of said fields with a second data source; and

performing a logical operation using said first and second data sources as operands, said logical operation being specified by said op-code.

26. (Previously Presented) The method of Claim 1, further comprising generating a long immediate constant using a single word instruction according to the method comprising:

providing an instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits;

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

27. (Previously Presented) The method of Claim 25, wherein said plurality of instruction words further comprises at least one instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits, said at least one instruction word with short immediate value being used to generate a long immediate constant according to the method comprising:

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

28. (Previously Presented) The method of Claim 27, wherein said at least one instruction word having a plurality of fields and said at least one instruction word having a short immediate value comprise the same instruction word(s).

29. (Previously Presented) The digital processor of Claim 14, wherein said at least one of said plurality of instruction words comprises an op-code and a plurality of fields, each of said fields comprising a plurality of bits, said at least one instruction word being encoded by:

associating a first of said fields with a first data source;

associating a second of said fields with a second data source; and

performing a logical operation using said first and second data sources as operands, said logical operation being specified by said op-code.

30. (Previously Presented) The digital processor of Claim 14, wherein said processor is further adapted to generate a long immediate constant using a single word instruction by:

providing an instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits;

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

31. (Previously Presented) The digital processor of Claim 29, wherein said plurality of instruction words further comprises at least one instruction word having an op-code and at least one short immediate value associated therewith, said at least one short immediate value comprising a plurality of bits, said at least one instruction word with short immediate value being used to generate a long immediate constant according to the method comprising:

selecting a portion of said plurality of bits of said at least one short immediate value;

shifting all of said bits of said at least one short immediate value using said op-code and only said portion of bits to produce a shifted immediate value; and

storing said shifted immediate value in a register.

32. (Previously Presented) The digital processor of Claim 31, wherein said at least one instruction word having a plurality of fields and said at least one instruction word having a short immediate value comprise the same instruction word(s).

33. (Cancelled)

34. (Cancelled)

35. (Previously Presented) A method of controlling the execution of instructions within a pipelined processor, comprising:

providing an instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said words comprising a jump instruction;

assigning one of a plurality of values to first and second of said data bits of said at least one jump instruction, said first and second bits adapted to define four discrete jump modes, said four discrete jump modes including one user-defined jump mode; and

controlling the execution of at least one subsequent instruction within said pipeline based on said assigned values of said first and second data bits when said at least one jump instruction is decoded.

36. (Cancelled)

37. (Previously Presented) An extensible pipelined digital processor having an instruction set comprising a plurality of basecase instructions and at least one extension instruction, at least one of said basecase and extension instructions comprising a branch instruction having at least one user-configurable mode and a plurality of other modes controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline.

38. (Previously Presented) An extensible pipelined digital processor having an instruction set comprising a plurality of basecase instructions and at least one extension instruction, at least one of said basecase and extension instructions comprising a branch instruction including two data bits defining four discrete modes controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline.

39. (Previously Presented) An extensible pipelined digital processor having an instruction set, said processor comprising:

a basecase processor core configuration including a base instruction set; and

at least one user-configured extension instruction within said instruction set, said at least one extension instruction comprising a branch instruction having at least one user-defined mode and a plurality of other modes controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline.

40. (Previously Presented) An extensible pipelined digital processor having basecase and extension instruction sets, at least one instruction within said basecase set comprising a branch instruction having at least four discrete modes for controlling the execution of at least one instruction in a delay slot following said branch instruction within said pipeline, said processor comprising:

a basecase processor core configuration including said base instruction set; and

at least one user-customized extension instruction within said instruction set.

41. (Currently Amended) A digital processor having at least one pipeline and an associated data storage device containing at least a portion of an instruction set comprising a plurality of instruction words, each of said instruction words comprising a plurality of data bits, at least one of said words comprising a branch instruction, wherein the execution of instructions within said at least one pipeline is controlled by (i) the assignment of one of a plurality of values to at least first, second, and third of said data bits of said at least one branch instruction, said first, second, and third bits adapted to define at least four discrete branch modes; and (ii) the execution of at least one subsequent instruction within said pipeline based on said assigned values of said first, second, and third data bits, when said at least one branch instruction is decoded, wherein said branch instruction is further configured to permit the definition of at least one branch or non-branch mode by a user using said at least first, second, and third bits.

42. (Previously Presented) The digital processor of Claim 41, wherein first and second of said at least four branch modes implement one- and two-cycle stalls within said pipeline, respectively.

43. (Previously Presented) The digital processor of Claim 41, wherein at least one of said at least four branch modes comprises a user-configurable mode.